

Docket No.: P-212

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of :  
:

Sang Ho LEE and Jang Keun OH :  
:

Serial No.: New U.S. Patent Application :  
:

Filed: August 16, 2001 :  
:

For: METHOD FOR MEASURING QUANTITY OF USAGE OF CPU

J1046 U.S. PTO  
09/930447  
08/16/01

#2  
D. Scott  
121481

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents  
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the following application:

Korean Patent Application No. 50037, filed August 28, 2000

A copy of each priority application listed above is enclosed.

Respectfully submitted,  
FLESHNER & KIM, LLP

Daniel Y.J. Kim  
Registration No. 36,186

P. O. Box 221200  
Chantilly, Virginia 20153-1200  
703 502-9440

Date: August 16, 2001

DYK/jld

J1046 U.S. PTO  
09/930447  
08/16/01



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Industrial  
Property Office.

출 원 번 호 : 특허출원 2000년 제 50037 호  
Application Number

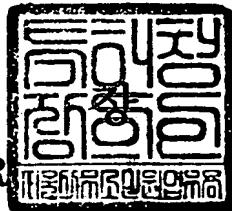
출 원 년 월 일 : 2000년 08월 28일  
Date of Application

출 원 인 : 엘지전자 주식회사  
Applicant(s)

2000 년 10 월 19 일

특 허 청

COMMISSIONER



【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0005
【제출일자】	2000.08.28
【국제특허분류】	G06F 1/00
【발명의 명칭】	씨피유 사용량 측정 방법
【발명의 영문명칭】	MEASURING METHOD FOR CPU USAGE
【출원인】	
【명칭】	엘지전자 주식회사
【출원인코드】	1-1998-000275-8
【대리인】	
【성명】	박장원
【대리인코드】	9-1998-000202-3
【포괄위임등록번호】	2000-027763-7
【발명자】	
【성명의 국문표기】	이상호
【성명의 영문표기】	LEE, Sang Ho
【주민등록번호】	750204-1691720
【우편번호】	136-075
【주소】	서울특별시 성북구 안암동5가 93-4
【국적】	KR
【발명자】	
【성명의 국문표기】	오장근
【성명의 영문표기】	OH, Jang Keun
【주민등록번호】	690520-1406416
【우편번호】	441-390
【주소】	경기도 수원시 권선구 권선동 신우아파트 708-1105
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대 리인 원 (인) <span style="float: right;">박장</span>

1020000050037

2000/10/2

【수수료】

【기본출원료】	12	면	29,000 원
【가산출원료】	0	면	0 원
【우선권주장료】	0	건	0 원
【심사청구료】	0	항	0 원
【합계】		29,000 원	
【첨부서류】		1. 요약서·명세서(도면)_1통	

**【요약서】****【요약】**

본 발명은 씨피유 사용량 측정 방법에 관한 것으로, 운영 시스템에서 제공하는 각종 함수들을 이용하여 소정 타이머 시간 간격으로 시스템 쓰레드를 제외한 다른 모든 쓰레드의 수행시간을 읽어 그 값을 가산하는 과정과; 상기 쓰레드 수행시간의 총합에서 이전에 수행되어 저장되어 있던 쓰레드 수행시간의 총합을 감산하는 과정과; 상기 감산된 쓰레드 수행시간을 상기 소정 타이머 시간으로 제산하여 씨피유 사용량을 산출하는 과정에 의해 보다 신뢰성 있는 씨피유 사용량을 직접 얻을 수 있도록 하는 효과가 있다.

**【대표도】**

도 1

## 【명세서】

### 【발명의 명칭】

씨피유 사용량 측정 방법{MEASURING METHOD FOR CPU USAGE}

### 【도면의 간단한 설명】

도1은 본 발명에 의해 씨피유 사용량을 측정하기 위한 상세 흐름도.

도2는 상기 도1에서 씨피유 사용량을 계산하는 프로시져와 외부 프로그램간의 인터페이스를 나타낸 예시도.

### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <3> 본 발명은 씨피유 사용량 측정 방법에 관한 것으로, 특히 운영 시스템에서 제공하는 각종 서비스들을 이용하여 보다 신뢰성 있는 씨피유 사용량을 직접 얻을 수 있도록 하는 씨피유 사용량 측정 방법에 관한 것이다.
- <4> 종래 기술의 시스템 사용량을 예측하기 위한 알고리즘은 다양한 종류가 있으며, 복잡한 알고리즘일수록 그 데이터의 신뢰도는 크며, 간단할수록 신뢰도가 떨어짐은 쉽게 예측 가능하다.
- <5> 또한, 운영 시스템(Operating System)에 따라 씨피유의 사용량을 계산하는 방식에는 차이가 발생할 수 있다.
- <6> 예를 들면, MS-Windows 9x는 운영 시스템에서 시스템이 유휴 상태일 경우, 각 디바이스 드라이버에 시스템 아이들(SYSTEM\_IDLE) 메시지를 전달한다. 즉, 하나의 드라이버에서

상기 시스템 아이들(SYSTEM\_IDLE) 메시지를 처리하는 프로시저를 두고, 메시지에 의해 프로시저가 호출(Call)될 때마다 카운터를 증가시키도록 한다면, 일정 시간동안 시스템 사용량이 적을 때는 카운터의 증가가 크다. 반면, 시스템의 사용량이 큰 경우 메시지의 생성빈도가 작아지므로 카운터의 증가가 작다.

<7> 또한, MS-Windows 9x 시스템에서는 자체적인 씨피유 사용량을 계산하는 알고리즘을 이용하여, 레지스트리(Registry)에 관련 정보를 업데이트시키고 있으며, 응용 프로그램(Application Program)을 개발할 때에는 그 레지스트리 정보를 이용할 것을 추천하고 있다.

<8> 그러나, 씨피유 사용량 계산을 위한 알고리즘이 명확하지 않으며, 시스템이 작업을 수행하고 있지 않음에도 불구하고 씨피유 사용량이 50% 이상의 값을 보이는 문제점이 있었다.

<9> 또한, 알고리즘이 명확하지 않음으로 인해, 씨피유 사용량을 업데이트하는 시간 간격에 대한 정보도 없으며, 이에 따라 짧은 순간의 씨피유 사용량을 판단하기에도 적절하지 못한 문제점이 있었다.

#### 【발명이 이루고자 하는 기술적 과제】

<10> 따라서, 본 발명은 상기와 같은 종래의 문제점을 해결하기 위하여 창출한 것으로, 씨피유 사용량이 저장되는 레지스트리 대신에 운영 시스템에서 제공하는 각종 함수들을 이용하여 보다 신뢰성 있는 씨피유 사용량을 직접 얻을 수 있도록 하는 씨피유 사용량 측정 방법을 제공함에 그 목적이 있다.

### 【발명의 구성 및 작용】

<11> 이와 같은 목적을 달성하기 위한 본 발명은, 소정시간 간격으로 시스템 쓰레드를 제외한 다른 모든 쓰레드 핸들을 읽어 그 값을 가산하는 과정과; 상기 쓰레드 핸들의 총합에서 이전에 수행되어 저장되어 있던 쓰레드 핸들의 총합을 감산하는 과정과; 상기 감산된 쓰레드 핸들을 상기 소정시간으로 제산하여 씨피유 사용량을 산출하는 과정으로 이루어진 것을 특징으로 한다.

<12> 이하, 본 발명에 따른 일실시예를 첨부한 도면을 참조하여 상세히 설명하면 다음과 같다

<13> 도1은 본 발명에 의해 씨피유 사용량을 측정하기 위한 상세 흐름도로서, 윈도우즈(MS-Windows)에서 제공하는 시스템 서비스를 이용하여 씨피유 사용량을 계산한다.

<14> 보다 정확하게는, 윈도우즈에서 가장 디바이스 드라이버(VXD)를 구현할 때 사용할 수 있는 브이엠엠(Virtual Machine Manager, VMM) 서비스를 이용하는 것으로, 그 중에서도 타이머 및 쓰레드에 관련된 서비스를 이용하는 것이다.

<15> 이와 같이, 브이엠엠 서비스를 이용하는 것은 로우 레벨(Low-level)보다 더 나은 기능을 수행할 수 있기 때문이다.

<16> 일반적으로, 시스템에서 동작하는 프로그램은 쓰레드(Thread)로 분류되며, 상기 쓰레드가 실행의 최소단위가 된다. 여기서, 실행이란 씨피유가 사용되는 것을 의미한다.

<17> 이에 도시한 바와 같이, 씨피유의 사용량을 계산하기 위하여 프로시저로 전환하면, 첫 번째 쓰레드(시스템 쓰레드)를 제외한 다른 모든 쓰레드의 수행시간의 총 합을 저장하기 위한 변수(SUM)를 0으로 초기화 시킨다(S80).

- <18> 이후, 상기 첫 번째 쓰레드(시스템 쓰레드)의 포인터 역할을 하는 핸들(Hd1)값을 읽은 후(S81), 그 다음 쓰레드의 핸들(Hd2)값을 읽는다(S82).
- <19> 다음, 상기 첫 번째 쓰레드(시스템 쓰레드)의 핸들값과 그 다음 쓰레드의 핸들값이 서로 동일한 값을 갖는지 비교한다(S83).
- <20> 즉, 상기 순차로 읽은 쓰레드의 핸들값이 시스템 쓰레드의 핸들과 동일한지를 비교함으로써, 전체 쓰레드를 모두 읽었는지 판단하는 것이다.
- <21> 여기서, 상기 각 쓰레드 핸들은 운영 시스템의 조정(Scheduling)에 의해서 순차적으로 반복하여 실행되고, 운영 시스템은 그 쓰레드 및 쓰레드 실행에 관한 정보를 관리한다.
- <22> 그리고, 상기 쓰레드 및 그 쓰레드 실행에 관한 정보를 브이엠엠 서비스에 의해 알 수 있는 것이다.
- <23> 이러한, 브이엠엠 서비스에는 첫 번째 쓰레드(시스템 쓰레드)의 핸들을 얻는 함수(Get\_Sys\_Thread\_Handle), 다음 쓰레드의 핸들을 얻는 함수(Get\_Next\_Thread\_Handle) 및 쓰레드 생성 후 부터의 실행 시간을 얻는 함수(\_GetThreadExecTime) 등이 포함된다.
- <24> 이후, 상기 쓰레드 핸들의 비교 과정(S83)에서, 첫 번째 쓰레드(시스템 쓰레드)의 핸들값과 다음 순차로 읽는 쓰레드의 핸들값이 다르면, 그 다음 쓰레드의 수행시간을 읽는다(S84).
- <25> 이후, 상기 읽은 쓰레드의 수행시간을 쓰레드의 총 수행시간을 저장하는 변수(SUM)에 더 해준다(S85).
- <26> 이 과정은 첫 번째 쓰레드(시스템 쓰레드)가 다시 읽혀질 때 까지 반복 수행 된다.
- <27> 이와 같이, 전체 쓰레드를 모두 읽으면, 상기 시스템 쓰레드를 제외한 다른 모든 쓰레드

의 수행시간의 총합이 변수(SUM)에 저장된다.

<28> 따라서, 상기 변수(SUM)에 저장된 쓰레드의 수행시간(SUM\_NOW)에서 그 이전에 저장된 쓰레드들의 수행시간(SUM\_PREV)을 뺀 후, 다시 그 값을 타이머 시간(TIMER)으로 나누면 (S86) 씨피유 사용량(CPU\_USAGE)이 측정되는 것이다.

<29> 예를 들면, 상기 초기에 저장된 쓰레드의 수행시간이 100초이고, 10초(타이머 시간)가 지난 후의 현재 수행된 쓰레드의 시간이 105초인 경우에, 상기 초기에 저장된 수행시간(100초)을 뺀 나머지 시간(5초)을 타이머 시간(10초)으로 나눈값이 씨피유의 사용량이 되는 것이다.

<30> 즉, 브이엠엠 서비스를 이용하여 일정시간(타이머 시간)마다 시스템 쓰레드를 제외한 나머지 쓰레드들의 수행시간의 총합을 계산한 후, 이전에 계산된 총 수행시간을 빼면, 상기 타이머 시간동안의 수행시간을 얻을 수 있으며, 이를 다시 상기 타이머 시간으로 나누어 주면 시스템의 사용량을 측정할 수 있다.

<31> 따라서, 만약 상기 타이머 시간을 짧게 설정한 경우에, 상기 계산된 씨피유 사용량은 그 타이머의 시간을 길게 할 경우보다 급격한 변화를 보이기 때문에, 이러한 급격한 변화를 줄이기 위해 이전에 계산된 씨피유 사용량들(CPU\_USAGE\_PREV, CPU\_USAGE\_PREV\_PREV)과 현재 계산된 씨피유 사용량(CPU\_USAGE\_NOW)을 이용하여 그 평균을 내어 사용량을 보정한다(S88).

<32> 즉, 타이머에 설정된 시간이 짧은 경우, 상기 계산된 씨피유의 사용량은 급격한 변화를 보이며, 씨피유 사용량에 따라 민감하게 반응하는 어플리케이션 구성을 할 때도 급격한 변화를 보일 수 있다.

- <33> 따라서, 상기 계산된 씨피유 사용량을 보정해주어 이러한 현상을 피할 수 있다.
- <34> 만약, 상기 타이머의 시간이 충분히 길 경우에는, 상기 계산된 씨피유의 사용량을 그대로 사용한다(S89).
- <35> 이후, 상기 씨피유 사용량이 측정되면, 상기 타이머를 예약하여 소정의 시간후에 다시 프로시저에 진입하도록 설정한 후, 전체 프로시저를 끝마치게 된다(S90).
- <36> 도2는 상기 도1에서 씨피유 사용량을 계산하는 프로시저와 외부 프로그램간의 인터페이스를 나타낸 예시도로서, 도1의 알고리즘에 의해 계산된 씨피유 사용량을 레지스트리에 업데이트시키거나 응용 프로그램(Win App)과 디바이스 드라이버(VxD)간의 인터페이스를 통해 윈도우 프로그램에서 액세스하도록 할 수 있으며, 다른 디바이스 드라이버에서 이용하도록 구현할 수도 있다.
- <37> 이에 도시한 바와 같이, 도1에서 계산된 씨피유의 사용량은 메모리에 지속적으로 업데이트 된다.
- <38> 다른, 디바이스 드라이버(VxD)에서는 씨피유 사용량을 디바이스 드라이버간의 인터페이싱(Interfacing) 방식을 통하여 얻을 수 있다.
- <39> 또한, 윈도우 어플리케이션으로 씨피유의 사용량을 전달하는 방법으로는 윈도우 레지스트리를 업데이트시키고, 응용 프로그램에서 이를 읽어가도록 하는 방법이 있다.
- <40> 또한, 상기 응용 프로그램과 디바이스 입출력 제어(Device I/O Control)를 이용하여 직접적으로 씨피유의 사용량을 읽어오는 방법을 사용할 수 있다.

### 【발명의 효과】

- <41> 이상에서 설명한 바와 같이 본 발명 씨피유 사용량 측정 방법은 운영 시스템에서 제공하

1020000050037

2000/10/2

는 각종 함수들을 이용하여 보다 신뢰성 있는 씨피유 사용량을 얻을 수 있는 효과가 있다.

### 【특허청구범위】

#### 【청구항 1】

소정 타이머 시간 간격으로 시스템 쓰레드를 제외한 다른 모든 쓰레드의 수행시간을 읽어 그 값을 가산하는 과정과; 상기 쓰레드 수행시간의 총합에서 이전에 수행되어 저장되어 있던 쓰레드 수행시간의 총합을 감산하는 과정과; 상기 감산된 쓰레드 수행시간을 상기 소정 타이머 시간으로 제산하여 씨피유 사용량을 산출하는 과정으로 이루어지는 것을 특징으로 하는 씨피유 사용량 측정 방법.

#### 【청구항 2】

제1항에 있어서, 상기 쓰레드 수행시간의 총합을 구하기 위한 과정은, 시스템 쓰레드를 시작으로 연속되는 다음 쓰레드의 핸들을 순차로 읽고, 그 값이 상기 시스템 쓰레드의 핸들과 같은지를 비교하여, 같은 값이 읽혀질 때 까지 다음 쓰레드의 수행시간 가산하는 과정을 반복 수행하여 이루어지는 것을 특징으로 하는 씨피유 사용량 측정 방법.

#### 【청구항 3】

제1항에 있어서, 상기 씨피유 사용량을 측정하기 위한 소정의 타이머 시간은 그 간격이 작을 경우, 급격한 사용량값 변화를 줄이기 위해 이전의 씨피유 사용량들과 현재 측정된 씨피유 사용량을 평균등에 의해 보정할 수 있도록 이루어진 것을 특징으로 하는 씨피유 사용량 측정 방법.

#### 【청구항 4】

제1항에 있어서, 상기 씨피유 사용량을 측정하기 위한 소정의 타이머 시간은 그 간격이

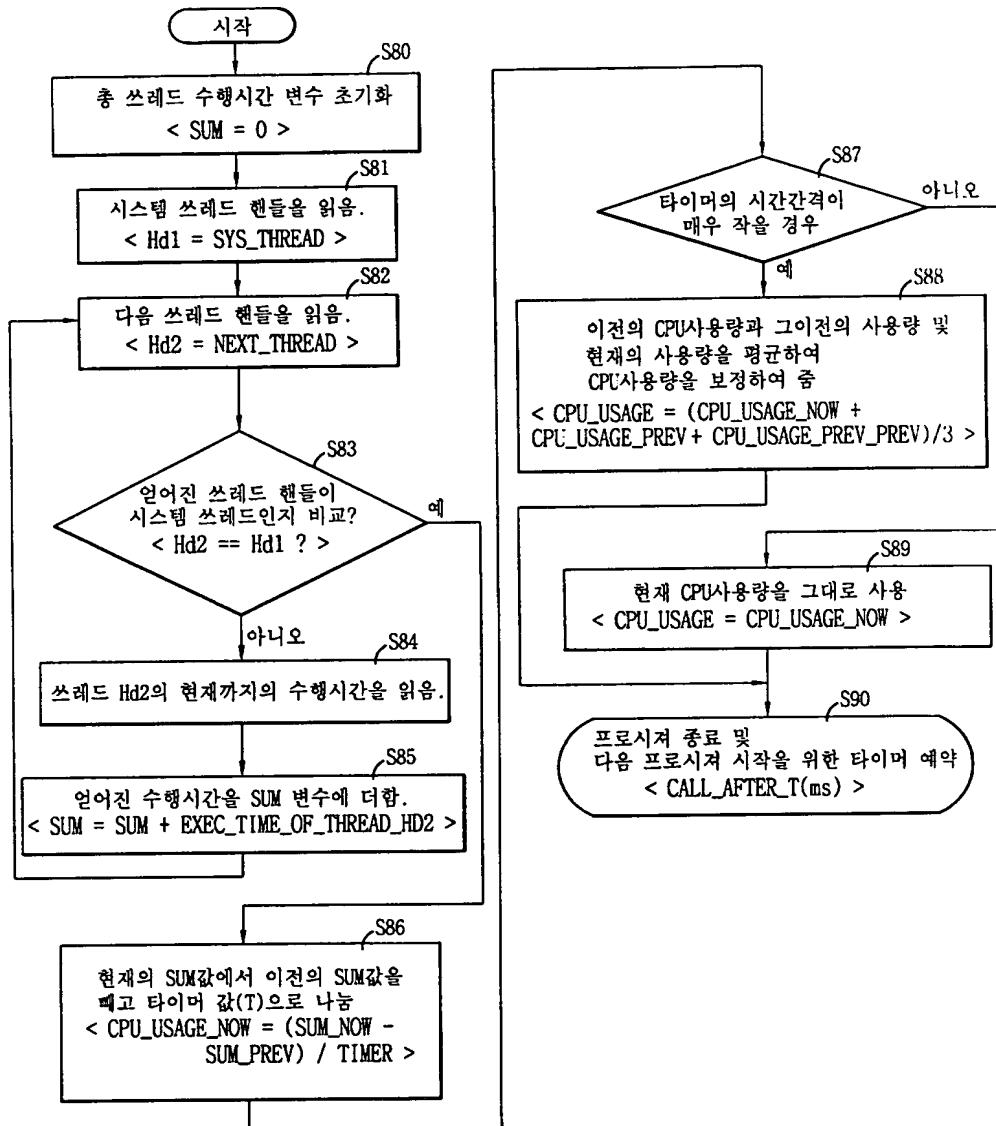
충분히 클 경우, 현재 측정된 씨피유 사용량을 그대로 사용할 수 있도록 이루어진 것을 특징으로 하는 씨피유 사용량 측정 방법.

#### 【청구항 5】

제1항 또는 제3항에 있어서, 소정의 타이머 시간 간격이 작아 씨피유 사용량이 급격한 변화를 보일 경우, 사용량 값을 그대로 이용하고자 할 경우는 현재 측정된 씨피유 사용량을 그대로 사용할 수 있도록 이루어진 것을 특징으로 하는 씨피유 사용량 측정 방법.

## 【도면】

【도 1】



## 【도 2】

